





Orbit'in Süper Fonksiyonu! 🔗





Senaryo

Orbit uzaya çıkmadan önce ısınma alıştırması yapıyor 🚀

Görev basit: 5 sayı say ve her birinde kendini hazırlamaya devam et!

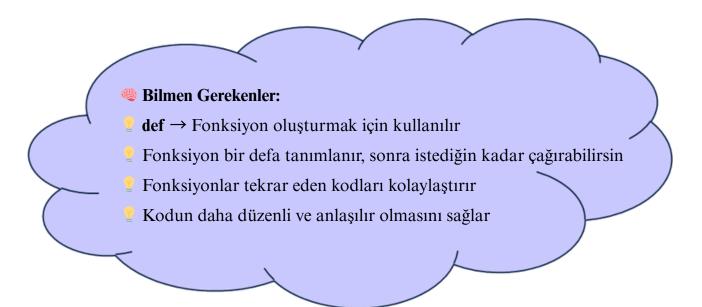
Bu sadece bir ısınma ama çok önemli! 🍑 🌣



- 📌 Orbit için bir fonksiyon oluştur
- 📌 Fonksiyon çalışınca "Orbit: Hazırım, göreve başlıyorum!" yazsın 🚀
- ★ Fonksiyonu en az bir kez çağır (çalıştır)
- 🖈 Fonksiyonlar neden kullanılır, düşün!

Uygulama-1

Function



```
def orbiti_baslat():
    print("% Orbit: Hazırım, göreve başlıyorum! %")
# Fonksiyonu çağırıyoruz
orbiti_baslat()
```

*

- ✓ İlk fonksiyonumu yazdım
- ✓ Fonksiyon tanımlamayı ve çağırmayı öğrendim
- ✓ Kodumu tekrar kullanabilir hale getirdim
- 🗸 Orbit'in sabah sistemini ben başlattım! 🔋 🖭







Orbit'in Sayı Dedektifi Görevi







Senaryo

Orbit bugün Sayı Dedektifi oldu! 🕶 🔢

Verilen herhangi bir sayının tek mi çift mi olduğunu anında tespit ediyor.

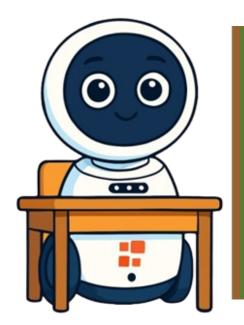
Ama bunu her seferinde tekrar yazmak yerine bir fonksiyon tanımlayıp işini kolaylaştırıyor.

Bakalım senin verdiğin sayı tek mi, çift mi? 👺



- 📌 Bir sayı al ve bu sayının çift mi tek mi olduğunu kontrol et 🔢
- 📌 Fonksiyon tanımlayarak bu kontrolü yap.
- 🖈 Sayıya göre uygun mesajı yazdır.
- 📌 Kodun düzenli ve tekrar kullanılabilir olmasını sağla.





- **Bilmen Gerekenler:**
- Fonksiyonlara bilgi (parametre) verebiliriz
- \P % (modül) operatörü kalanı bulur: sayı % 2 == 0 → çift sayı
- Fonksiyonlar sayesinde kod tekrarını önleriz
- P Birden fazla sayıyı kolayca kontrol edebiliriz

```
# Fonksiyon tanım1
def cift_mi_tek_mi(sayi):
    if sayi % 2 == 0:
        print(f"Q {sayi} → Cift say1! ✓")
    else:
        print(f"Q {sayi} → Tek say1! ✓")

# Fonksiyonu kullanma
cift_mi_tek_mi (7)
cift_mi_tek_mi (12)
```

*

- 🗸 Bir fonksiyon içinde mantıksal kontrol yapmayı öğrendim 🧠
- Sayıların özelliklerini analiz etmeyi pekiştirdim 🔍
- ✓ Orbit'in sayı dedektifi görevini başarıyla tamamladım! ≫≅







Orbit'in Mini Hesap Makinesi Görevi <u>■</u> •••





Senaryo

Orbit bugün Sayı Dedektifi oldu! 🕶 🔢

Verilen herhangi bir sayının tek mi çift mi olduğunu anında tespit ediyor.

Ama bunu her seferinde tekrar yazmak yerine bir fonksiyon tanımlayıp işini kolaylaştırıyor.

Bakalım senin verdiğin sayı tek mi, çift mi? 🤔

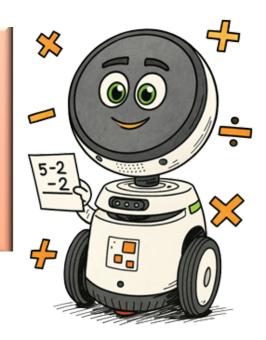


- 📌 Bir sayı al ve bu sayının çift mi tek mi olduğunu kontrol et 🔢
- 📌 Fonksiyon tanımlayarak bu kontrolü yap.
- 🖈 Sayıya göre uygun mesajı yazdır.
- 📌 Kodun düzenli ve tekrar kullanılabilir olmasını sağla.





- **Bilmen Gerekenler:**
- def ile yeni bir fonksiyon oluşturulur
- 🛂 a ve b parametrelerdir: fonksiyon bu sayılarla işlem yapar
- return sonucu dışarı iletir, böylece başka yerde kullanılabilir
- Fonksiyonlar işi bölerek kodu temiz ve tekrar kullanılabilir yapar



```
# 1) Toplama fonksiyonu
def topla(a, b):
   return a + b
# 2) Çıkarma fonksiyonu
def cikar(a, b):
   return a - b
# 3) Çarpma fonksiyonu
def carp(a, b):
   return a * b
# 4) Bölme fonksiyonu
def bol(a, b):
   return a / b
# Örnek kullanımlar
sayi1 = 15
sayi2 = 4
print("+", sayi1, "+", sayi2, "=", topla(sayi1, sayi2))
print("-", sayi1, "-", sayi2, "=", cikar(sayi1, sayi2))
print("X", sayi1, "x", sayi2, "=", carp(sayi1, sayi2))
print("+", sayi1, "/", sayi2, "=", bol(sayi1, sayi2))
```



Uygulama-3

Function



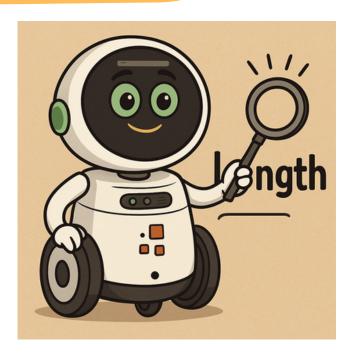
- ✓ Dört temel işlemi fonksiyon yazarak yaptım
- ✓ Parametre gönderip sonucu aldım
- ✓ return ile sonuçları başka işlemlerde kullanabildim
- ✓ Kodun düzenli olması için tekrar eden kodları fonksiyona aktardım
- ✓ Orbit'in hesap makinesi görevi başarıyla tamamlandı! ※







Orbit'in Kelime Dedektörü 🔍





Senaryo

Bugün Orbit'in görevi, insanların söyledikleri kelimeleri analiz etmek! 🕮 ...

Ama bu sıradan bir analiz değil...

Her kelimenin içinde ne kadar enerji taşıdığını harf sayısına göre ölçüyor! 🌟

Orbit'in dedektör sistemi kelimenin uzunluğunu tespit ediyor, sonra da sonucu bildiriyor.

Bakalım senin yazdığın kelime kaç enerji birimi taşıyor? 🗲 🦠



- 📌 Kullanıcıdan bir kelime al 📝
- 📌 Kelimeyi Orbit'in analizine gönder 📚
- 🖈 Kelimenin harf sayısını ölçen bir fonksiyon yaz 羞
- 📌 Orbit sonucu ekrana eğlenceli bir mesajla yazsın! 🖶
- 📌 Farklı kelimelerle deneyerek dedektörü test et! 🧳
- 📌 Orbit'in "enerji dedektörü" temasıyla hayal gücünü kat! 🤩





- **Bilmen Gerekenler:**
- def → Yeni bir fonksiyon tanımlamak için kullanılır
- return → Fonksiyonun dışına bir değer döndürmek için
 kullanılır
- Fonksiyonlar, bir işlemi tekrar kullanmak için kodumuzu sadeleştirir
- input() → Kullanıcıdan bilgi almak için kullanılır



```
# Kelimenin uzunluğunu ölçen özel dedektör fonksiyonu

def kelime_uzunlugu(kelime):
    return len(kelime)

# Kullanıcıdan kelime alınır
girdi = input(" Orbit için bir kelime yaz (örneğin: 'gezegen'): ")

# Orbit kelimeyi analiz ediyor
uzunluk = kelime_uzunlugu(girdi)

# Sonuç ekranı
print(" Orbit: Enerji analiz tamamlandı!")
print(f" '{girdi}' kelimesi {uzunluk} harf içeriyor. Harika bir enerji seviyesi! 英")
```



Uygulama-4

Function



- ✓ Bir fonksiyon yazıp çağırmayı öğrendim
- ✓len() fonksiyonu ile metinlerin uzunluğunu hesapladım
- ✓ return komutu ile fonksiyonun sonucunu kullanmayı öğrendim
- ✓ Kullanıcıdan bilgi alıp bu bilgiyi işledim
- ✓ Orbit'in kelime analiz robotuna enerji verdim! 🗓 🖭





Orbit'in Akıllı Hesaplayıcısı 🥮





Senaryo

Orbit'e bir görev verildi:

Kullanıcıdan iki sayı alacak ve hangi işlemi yapmak istediğini soracak.

Kullanıcı "toplama", "çıkarma", "çarpma" veya "bölme" seçeneklerinden birini yazarsa, Orbit hemen sonucu hesaplayacak!

Fonksiyonlar sayesinde her işlem temiz bir şekilde gerçekleşecek.

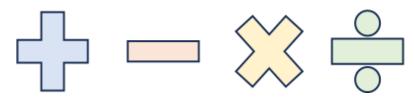


- 📌 Kullanıcıdan iki sayı al 🔢
- 📌 Kullanıcıdan işlem türünü iste (topla, çıkar, çarp, böl) 🚆
- 📌 Dört işlem için ayrı fonksiyonlar oluştur
- 🖈 Seçilen işleme göre doğru fonksiyonu çağır
- 📌 Sonucu ekrana yazdır 🖶









- 66
 - **Bilmen Gerekenler:**
 - input() → Kullanıcıdan bilgi alır
 - 💡 if yapısı → Hangi işlemin seçildiğini kontrol eder
 - Her işlem için ayrı fonksiyon kod tekrarını engeller
 - 💡 Sıfıra bölme özel olarak kontrol edilmelidir! 🗘

```
# İşlem fonksiyonları
def topla(a, b):
    return a + b
def cikar(a, b):
    return a - b
def carp(a, b):
    return a * b
def bol(a, b):
    if b != 0:
        return a / b
    else:
        return "! Sıfıra bölünemez!"
```



```
# Kullanıcıdan bilgi al
print(" 🖭 Orbit: Hangi işlemi yapmak istersin? (topla, çıkar, çarp,
böl)")
islem = input(" işlem türünü yaz: ")
sayi1 = float(input(" Birinci sayıyı gir: "))
sayi2 = float(input("! ikinci sayıyı gir: "))
# İşlem seçimi ve sonuç
if islem == "topla":
   print("✓ Sonuç:", topla(sayi1, sayi2))
elif islem == "cikar":
   print("✓ Sonuç:", cikar(sayi1, sayi2))
elif islem == "carp":
   print("✓ Sonuç:", carp(sayi1, sayi2))
elif islem == "böl":
   print("✓ Sonuç:", bol(sayi1, sayi2))
else:
   print("!Geçersiz işlem seçimi.")
```

- ✓ Fonksiyonlarla gerçek uygulama yazmayı öğrendim
- Kullanıcıdan veri alıp koşullarla yönlendirmeyi kavradım
- ▼ Kodun okunabilirliğini ve tekrar kullanılabilirliğini artırdım
- ✓ Orbit ile birlikte tam bir mini hesap makinesi tasarladım! 🕮 🔢